Decentralised trust-management inspired by ant pheromones

Decentralised Trust-Management Inspired By Ant Pheromones

Sarah Edenhofer[1], Sven Tomforde[2], Darius Fischer[1], Jörg Hähner[1], Florian Menzel[3], Sebastian von Mammen[1]

[1]Organic Computing Group, University of Augsburg, Eichleitnerstr. 30, 86159 Augsburg, Germany, E-Mail: {<name>.<surname>}@informatik.uni-augsburg.de

[2]Intelligent Embedded Systems Group, University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany, E-Mail: stomforde@uni-kassel.de

[3]Department of Evolutionary Biology, Institute of Zoology, University of Mainz, Johannes-von-Müller-Weg 6, 55128 Mainz, Germany, E-Mail: menzelf@uni-mainz.de

Abstract: Computational trust is increasingly utilised to select interaction partners in open technical systems consisting of heterogeneous, autonomous agents. Current approaches rely on centralised elements for managing trust ratings (i.e. control and provide access to aggregated ratings). Consider a grid computing application as illustrating example: agents share their computing resources and cooperate in terms of processing computing jobs. These agents are free to join and leave, and they decide on their own with whom to interact. The impact of malicious or uncooperative agents can be countered by only cooperating with agents that have shown to be benevolent: trust relationships are established. Typically, this requires a centralised data-base storing information about past interactions and their outcome. In this article, we propose a novel, decentralised trust mechanism inspired by the nestmate recognition system in ants. More precisely, the concept of recognition pheromones, which stick to the agents and cannot be removed or counterfeit, is turned into algorithmic logic and interaction protocols. We demonstrate the potential benefit by using simulations of the grid scenario.

Keywords: Trust, Distributed Verification, Ant Algorithms, Pheromones, Organic Computing, Asymmetric Encryption, Multi-agent Systems, Computing Grid Systems

Reference to this paper should be made as follows: Edenhofer, S. and Tomforde, S. and Fischer, D. and Hähner, J. and Menzel, F. and von Mammen, S. (2016) 'Decentralised Trust-Management Inspired By Ant Pheromones', *International Journal of Mobile Network Design and Innovation*, Special Issue on Signal Processing, Security and Privacy for Mobile/Wireless and Computer Networks, Vol. x, No. x, pp.xxx–xxx.

Biographical notes:

Sarah Edenhofer is a researcher at the Organic Computing group of Augsburg University, Germany. She received her M. Sc. in Computer Science in 2014 from Augsburg University, Germany. Her research focus lies on the field of Organic Computing, Interactive Simulations, and trust in open distributed Multi-agent Systems.

Sven Tomforde is a senior researcher at the Intelligent Embedded Systems group of Kassel University, Germany. He received his Dr.-Ing. degree (PhD) in Computer Science in 2011 from Leibniz University of Hannover, Germany and his M. Sc. in Computer Science in 2007 from both, Leibniz University of Hannover, Germany and University of Bristol, UK. He works in the field of Organic Computing, self-organising systems, and applied machine learning techniques.

Darius Fischer is a student at the University of Edinburgh, Scotland. He received his B. Sc. in Computer Science in 2015 from Augsburg University, Germany. His research interests are data analytics, computer security, and applied machine learning for healthcare.

Jörg Hähner is full professor for Organic Computing at Augsburg University, Germany. He received his Diplom degree from Darmstadt University of Technology, Germany, in 2001 and the Dr. rer. nat. degree (PhD) from University of Stuttgart, Germany, in 2006, both in Computer Science. His research focuses on architectures and algorithms for self-organised and distributed systems (e.g. distributed smart camera systems, mobile ad-hoc and sensor networks, and global scale peer-to-peer systems).

Florian Menzel is assistant professor at the Institute of Zoology, University of Mainz. He received his PhD in 2009, and his Diplom degree in 2005, both in biology, from the University of Würzburg, for his research on associations between ant species in Southeast Asia's tropical rainforests. His research focuses on chemical communication within and between ant species, the evolution of insect communication signals, and the mechanisms that structure ant communities in Europe and the tropics.

Sebastian von Mammen is a senior researcher at the Organic Computing group of Augsburg University, Germany. He received his PhD in Computer Science in 2009 from University of Calgary, Canada and his Diplom degree in Computer Science in 2005 from University Erlangen-Nürnberg, Germany. His research interests are self-organising systems, especially in terms of (self-)adaptation, accessible modelling, and automated abstraction.

1 Introduction

In 1991, Marc Weiser formulated his vision of Ubiquitous Computing, see Weiser (1991): Individual devices such as personal computer will be replaced by "intelligent things" and these things support humans in an imperceptible manner. About two and a half decades later, this vision has become increasingly realistic: *Information and Communication Technology* (ICT) pervades every aspect of our daily lives, since it is embedded in the environments that we encounter on a daily basis. In contrast to Weiser' initial description, services are still directly accessible through devices such as smart phones, but we also observe trends towards automated machine interaction as e.g. formulated by the Internet-of-Things, see Greengard (2015). As a consequence, we can observe an increasing number of interconnected and autonomous systems that interact in a demand-oriented manner, cf. Tomforde et al. (2014). In order to cope with the arising complexity, the solutions are typically characterised by heterogeneous, self-adaptive sub-systems that are loosely coupled and cooperating in a dynamically changing system structure. A major drawback in such complex structures is that only limited access control can be applied—meaning that the system is in principle open for participants to freely join in and leave.

Openness, heterogeneity, and dynamics introduce uncertainty about the expected behaviour of the participating elements—we will refer to entities or elements within these interconnected system structures as *agents* throughout the article, since we assume technical solutions acting on behalf of a user, see Wooldridge (2001). One way to cope with this kind of uncertainty is to establish computational trust (see Castelfranchi & Falcone (2010) for a definition) relations among agents and incorporate this concept within the decision logic when choosing interaction partners.

Consider a Desktop Computing Grid as an exemplary application where aspects like openness and heterogeneity play an important role. In such a grid system, participants cooperate in terms of sharing their computing resources, cf. Choi et al. (2008). More precisely, they process computational jobs for each other following the concept of mutuality. In general, participation is only beneficial if agents behave cooperatively. As a consequence, uncooperative or even malicious agents have to be isolated to maintain the best possible utility for benevolent agents. This can be achieved by establishing trust relationships, i.e. estimations of how cooperative interaction partners will behave, see Castelfranchi & Falcone (2010). These trust estimations take own experiences into account as well as those other (reliable) interaction partners have made with a certain agent. We will refer to such a trust-based approach as Trusted Desktop Grid (TDG, cf. Bernard et al. (2010)) within this article and use it as application scenario for investigating decentralised trust and reputation schemes.

Especially when realising the underlying trust mechanism in a fully decentralised manner to avoid surveillance and central control while simultaneously taking experiences into account that other, trusted interaction partners made with a certain entity, novel challenges come into play. Most importantly, we have to guarantee the identity of sender and receiver as well as the correctness of trust values provided by others. The concept of trust among anonymous agents in a decentralised system resembles recognition and trust in ant colonies. Like the trusted interaction partners, ant colonies comprise hundreds to millions of workers who do not know each other individually. Mutual trust and, hence, cooperation, between ant workers is only achieved by a specific odour signature that is shared by all colony members. Individuals that do not possess this signature are attacked and not admitted to the ant nest. Indeed, there are "malicious agents" in the form of other insect species ("ant guests", such as other ant species, beetles, butterfly larvae, or silverfishes) who try to gain access to an ant colony and parasitise on their food sources, cf. Bagnères & Lorenzi (2010). Therefore, a nestmate recognition system is crucial to maintain the social integrity of the colony and avoid its exploitation by other insects, cf. Leonhardt et al. (2016). To this end, this article proposes a novel scheme for trust messages inspired by nestmate recognition in ants. We call this concept *Trusted Digital Pheromones* (TruDiPhe). The contribution of this article is two-fold: On the one hand, we present a novel mechanism to allow for a secure exchange of trust information within distributed agent societies following a public-key concept. On the other hand, we analyse the behaviour within simulations of the TDG by considering different attack levels. In this context, we refer to the participation of uncooperative or even malicious agents as attack, since it decreases the utility of benevolent agents.

The remainder of this article is organised as follows: Section 2 briefly discusses related work and highlights the need for a novel decentralised trust mechanism. Section 3 presents the application scenario: the Trusted Desktop Computing Grid. Afterwards, we explain our approach of trusted digital pheromones in Section 4. The developed techniques are evaluated in Section 5 using simulations of our application scenario. Finally, Section 6 summarises the article and gives an outlook to future work.

2 Related Work

In this section, we discuss related work concerning distributed trust management in open systems inspired by ant pheromones. First, we describe natural processes which inspired us, especially colony affiliation and nestmate recognition among ant colonies. Then trust in Multi-agent Systems in general is introduced, followed by attacks and security measures in trust management systems, as well as (trust in) Desktop Computing Grid

Systems in particular. We conclude the section with a discussion of appropriate performance metrics.

2.1 Nature as Inspiration

Taking ideas from nature and transferring them to computer science is a well-established approach and the origin for the Organic Computing initiative, cf. Müller-Schloer et al. (2011). Just as Bionics (Dickinson (1999)) has opened up a new field of research dedicated to the creation of machines inspired by nature, there are many fields in computer science trying to model natural behaviour in algorithms. Especially social insects, such as ants or bees, have attracted interest of many researchers: Individual insects follow simple rules, yet, as a group and with the help of a complex, but decentralised communication system, they can effectively organise concerted actions (Leonhardt et al. (2016)). This principle can be transferred to technical systems: each insect represents an individual part of a system, and when they work together, effects may occur which cannot fully be foreseen. This refers to the concept of emergence, cf. Mnif & Müller-Schloer (2006). An example of this is the system of chemical (odour) signatures, which are used by insects to encode a variety of information.

2.2 Chemical Communication and Nestmate Recognition in Ant Colonies

Ants communicate using chemical scents (Hölldobler & Wilson (2009)). For example, they lay out scent trails ("trail pheromones") which lead the other colony members to food sources, cf. Morgan (2009). These pheromones are chemical messages, which can be passed from one ant to the other without them meeting physically.

A different type of scents is used to distinguish nestmates from individuals of other nests (nestmate recognition). These "recognition pheromones" are cuticular hydrocarbons, which cover every ant's body surface and are little volatile. Every ant thus carries a cuticular hydrocarbon profile, which is perceived by every individual it encounters and cannot be hidden. It contains a variety of information, such as gender, species membership, and colony membership. Upon touching another individual with the antennae, ants can decode the information of the other's scent profile almost instantly. As a consequence, only members of the own colony are allowed into the nest, while alien intruders are attacked and displaced (Leonhardt et al. (2016)). Each ant produces its own scent, which is largely genetically determined. The composition of this chemical blend differs between different individuals even within the same ant species. The common colony odour, which is used for nestmate recognition, originates from mixing scent profiles between individuals. To this end, the ants groom each other. Thereby, they take up each other's and their own cuticular hydrocarbons, which are stored in a specialised gland (the postpharyngeal gland), mixed, and redistributed onto themselves and other ants when cleaning or touching them (d'Ettorre & Lenoir (2010), Leonhardt et al. (2016)). This nestmate recognition system forms the conceptual foundation for our trust management algorithm.

2.3 Trust in Multi-agent Systems

The concept of *trust in Multi-agent Systems* (MAS) (Wooldridge (2001), Ramchurn et al. (2004)) is well studied in fields such as economy, evolutionary biology, anthropology, and sociology (Mui (2002)). The term is typically defined as follows: "Trust is a belief an agent has that the other party will do what it says it will (being honest and reliable) or reciprocate (being reciprocative for the common good of both), given an opportunity to defect to get higher payoffs" (Ramchurn et al. (2004)) and "Without trust, the (human) society would cease to exist" (Vu (2010)). Trust enables us to interact with all other people, also those whom we have never met before. Every person is first of all concerned with his or her individual interests, but still expects certain behaviours by other people and relies on them. Therefore, trust is an essential basis for a functioning society (Luhmann (1979)). In contrast, the closely related term *Reputation* is usually seen as the combined trust value of many agents which are rating one single agent (Mui (2002), Aberer et al. (2006)).

2.4 Attacks and Security Measures

Several disadvantages come with the openness and anonymity of MAS, such as online communities, and the autonomy of agents in these system. For example, malicious agents can participate without other agents being aware of their existence. In general, for an agent to act maliciously and exploit the system, the benefit of an attack must outweigh the costs, and the pay-off must be higher than the potential punishment if the attack is monitored and sanctioned (assuming rational behaviour). If the system simply makes it unattractive for an attacker to exploit it, one speaks of a *soft security* approach. If this is not enough, we need *hard security mechanisms*, such as cryptography, to make the attacks harder and even impossible (Carrara & Hogben (2007), Vu (2010)).

2.5 Desktop Computing Grid Systems

Desktop Computing Grid Systems such as the TDG (Section 3) are used to share resources between multiple administrative authorities. One example for a peer-to-peer based system is the ShareGrid Project (Anglano et al. (2008)). A second approach is the Organic Grid, a peer-to-peer based approach with decentralised scheduling (Chakravarti et al. (2004)). Compared to our system, these approaches assume benevolence (Wang & Vassileva (2004)), i.e. that there are no malicious agents participating and misbehaving. Another approach is the

open source Berkeley Open Infrastructure for Network Computing Project (BOINC, Anderson & Fedak (2006)) which follows the same routine, creating several replicas over and over again, and looking for differences in the results. This ensues in a large amount of replicas and, in consequence, huge computing power overhead. XtremWeb (Fedak et al. (2001)) aims at setting up a global computing application and "harvest[ing] the idle time of Internet connected computers which may be widely distributed across the world, to run a very large and distributed application" with an ad-hoc verification process for participating computers. A panoramic view on computational trust in MAS can be found in Ramchurn et al. (2004), Castelfranchi & Falcone (2010), or Sabater & Sierra (2005).

2.6 Performance Metrics

In order to compare different approaches of reputation systems, the performance of those systems has to be measured. A useful metric should show the *direct benefit* gained by each or all agents, independent of reputation system details. Zacharia & Maes (2000) and Carbo et al. (2003) both use the development of reputation values of malicious agents over time. While the reputation obviously decreases in their evaluations, it is difficult to tell how good this decrease is compared to other systems. Multiple approaches were plotted in one diagram in Carbo et al. (2002), but it remains unclear *how much* the performance of the systems differs. Kamvar et al. (2003) test their system by plotting the *load distribution*. In a reputation system where users always choose peers with the highest reputation for an interaction, these highly rated peers receive almost all of the requests. Due to limitations in bandwidth or computing power, the favoured peers cannot serve all enquiries. This decreases their reputation, until only a few requesters remain and the reputation increases again. This oscillating behaviour means an unnecessary overhead to the system, since each rejection delays the successful processing of the request. The better the load is distributed, the less waiting time the agents have to endure. Therefore, equal load distribution is one indicator of an efficient system. Furthermore, the benefit for a single agent is not obvious, since it arises only indirectly through shorter waiting times, which means more work finished in the same time. The *interaction success (failure) rate* (Xiong & Liu (2004), similar to the *fraction of (in)authentic downloads* in Kamvar et al. (2003)), measures the percentage of all interactions with a positive (negative) outcome. Here, one value is measured independently of reputation system details, but still does not take the direct benefit into account, e.g. one single agent in a TDG system which successfully processes one single work unit at a time would have a perfect score regarding this metric. A metric satisfying all of our requirements, albeit limited to the area of TDG systems, is *speedup* (Klejnowski (2014)). It is defined as the relation between the time it would have taken a single agent to process all work by itself and the time it actually takes to process the work (units) distributedly.

3 Application Scenario: Trusted Desktop Grid

We consider an open, distributed Trusted Desktop Grid (TDG) as application scenario for the distributed verification of trust values. In this scenario, we use an open and heterogeneous MAS and we do not assume benevolence. The agents in the system cooperate to gain an advantage. The mechanism determining this cooperation is trust. Because of the openness of the system, different agents may try to exploit it. They may be uncooperative, malfunctioning or even malicious.

An agent, which acts on behalf of the user, is submitting *jobs* it wants to have calculated (Klejnowski (2014)). Each job is composed of several independently processable *work units* (WUs). The agents are expected to volunteer their machines as workers for other agents' WUs as well as to share their resources. If two agents have cooperated, they evaluate each other afterwards.

3.1 Agent Goal and Global Goal

The benefit of an agent can be measured by the *speedup* σ, informally speaking the benefit achieved by distributedly processing a task, compared to having to process all work on its own (in accordance with Klejnowski (2014)). An agent has multiple ratings with values between 0 and 1 that it gets from other agents. The amount of ratings stored in the queue at any time is limited. As a consequence, ratings are forgotten after some time. The global average of all ratings for one single agent is called *reputation*. For further details see Kantert et al. (2015). These ratings allow to estimate the future behaviour of an agent, based on its previous actions. In our system, agents get a higher rating, if they work for other agents and a lower rating, if they reject or cancel work requests, cf. Klejnowski (2014).

A job J is a set of WUs, which is released in time step t_J^{rel} and completed in t_J^{compl}, when the last WU is completed. This is why the speedup can only be determined after the last result has been returned to the submitter.

The speedup σ in Equation 1 is a metric known from multi-core systems. It is based on the assumption that parallelisation helps to process a task (i.e. a job) faster than processing it on a single core.

$$\sigma = \frac{\sum_J (t_{self}^{compl} - t_{self}^{rel})}{\sum_J (t_{dist}^{compl} - t_{dist}^{rel})} \tag{1}$$

In short, we can write $\sigma := \frac{t_{self}}{t_{dist}}$ with t_{self} being the time it would require an agent to process all WUs of a job without cooperation, i.e. sequentially. t_{dist} is the time it takes until all WUs are computed distributedly

and the last result is returned to the submitting agent. If no cooperation partners can be found, agents need to calculate their own WUs. This results in a speedup value equal to one. In general, we assume that agents behave selfishly and only cooperate if they can expect an advantage, i.e. $\sigma > 1$.

The global goal—also referred to as the system goal—is to enable and encourage agents to cooperate and thereby achieve the best average speedup possible. The systems' focus is *coordination*, i.e. shaping the environment in a way that allows for cooperation and, thereby, leads to optimising the global goal.

3.2 Worker and Submitter Component

Each agent is free do decide which agent it wants to pass WUs to or receive from. Therefore, every agent has a *submitter* and a *worker* component.

The *submitter component* is the scheduler of the agent and responsible for distributing WUs. If an agent receives a job J from the user consisting of multiple WUs, it creates a list of suited workers, i.e. workers it trusts. It then asks workers from this list to cooperate and calculate WUs, until either no more WU or no more workers are left. If all workers were asked but there are still unprocessed WUs remaining, the agent calculates them on its own.

The *worker component* decides whether an agent wants to work for a certain submitter. When the agent receives a request to process a WU, it calculates its expected reward for accepting and rejecting the WU. If the reward of accepting the WU prevails, the agent takes the WU and puts it in its own working queue, where the WU remains until the agent starts to process it, i.e. until the other WUs in the queue were processed. Afterwards, it transfers the result back to the submitter where the result is validated. A job is completed, if all WUs are returned to the submitter.

3.3 Agent Types

In the context of our TDG, agents can show malicious behaviour like returning wrong results, or refusing to work for other agents while submitting WUs to them. Behaviour like this can lead to a lower average speedup. In the following, we discuss different classes of stereotype agent behaviour that are considered within our system.

Adaptive Agents (ADA) are cooperative as long as their peers act benevolently and give honest ratings. They work for other agents of good reputation in the system. If the WU-queue of an ADA is saturated to the limit of its capacity, the agent may reject another WU. The trust concepts presented in this article focus on this group of agents.

Egoists (EGO) accept most work requests but return fake results with a certain probability (we use a value of 0.2 within our simulations), this wastes the time of other agents. The egoists make it necessary to validate returned, presumably completed jobs. This results in a drop of the average speedup.

Altruistic Agents accept every job, regardless of the circumstances or cooperation partners.

Sloppy Agents are cooperative but do only accept a certain percentage of WUs offered to them (Edenhofer et al. (2015)).

Dishonest Agents (D...), e.g. Dishonest Adaptive Agents (DADAs) or Dishonest Egoistic Agents (DEGOs), give false ratings with the aim to harm others and to improve their own reputation compared to other agents. In our system, dishonest ratings are modelled by inverting the actual rating, e.g. -1 instead of 1.

4 The TruDiPhe Approach

In Section 2.2, we introduced a way of disseminating the common scent inside an ant colony which can be compared to decentralised trust management: The odours of all ants in the colony are mixed up to one common colony scent. The smell acts like a token of trust which is passed around between members of the colony.

In the following, we will algorithmically model this kind of decentralised trust management. We start by introducing our novel approach to digital pheromones called *TruDiPhes* (Trusted Digital Pheromones). The term 'TruDiPhe' refers to both the concept and one single message, depending on (and being clear from) the context.

4.1 Definition of TruDiPhe

A TruDiPhe includes the following information about the trust transmitted: the creator x and the recipient y of the scent, a timestamp TS (of the creation), the $Type$ (describing the context of the trust value), and amount (trust value $TV \in \mathbb{N}$). A message TDP representing a TruDiPhe is described in Equation 2 (adapted from Singh & Liu (2003)):

$$TDP(x, y, TS, TV, Type) = \\ (P_x(B_y, TV, TS, Type), B_x) \tag{2}$$

P_z and B_z stand for the private and public key of an agent $z, z \in \{x, y\}$, $P_z(msg)$ or $B_z(msg)$ represent the message resulting from the encoding of the message msg with the private / public key of agent z (see Rouse (2016) for information about assymetric cryptography). We will only work with the 'generic' type in this article, as this is sufficient for the demonstration of the advantages of our approach. Every agent with a copy of the TruDiPhe is called its *owner*.

TruDiPhes have five main characteristics, which correspond to the decisive properties of recognition pheromones:

(1) They stick to a target and cannot be removed easily. Trust management is mainly used in open, distributed networks. Due to uncertainty about the

behaviour of the agents and their autonomy, only positive trust values about the agent are stored at the agent itself, so it has no advantage when removing a TruDiPhe. This is one important fact to guarantee *full* distributedness and highlights the analogy to ant pheromones: *If only positive ratings are used in the reputation system, the trust values can be* ***stored at their recipients instead of their creators.***

(2) They cannot be altered or counterfeited. This is guarantueed by the use of asymmetric cryptography. By encoding all information with the private key P_x, a TruDiPhe cannot be altered without destroying all information stored in it. It cannot be counterfeited without possessing the private key. From the encoded message itself it is impossible to tell who encoded the message. Therefore, the creator's public key is appended: any other agent can validate the authenticity of the message without having to know the creator and its public key in advance. If the encoded message or the public key is altered, the decoded message will not show the correct public key of the TruDiPhe's recipient.

(3) They are transferable to other agents. This is fulfilled, since a TruDiPhe is a simple message and can be passed freely in the network.

(4) Their intensity can vary. The creator of a message chooses the trust value freely, a greater value represents a higher intensity. In the TDG, the trust value could, for example, differ dependent on the work unit size, or the processing time.

(5) They evaporate over time. Since our TruDiPhes include a time stamp, this property is also satisfied. Even more, it gives each agent the possibility to choose from which period of time it wants to take odours into account for its considerations about work distribution. Furthermore, the timestamp helps to prevent the recipient from simply duplicating TruDiPhes for a higher, unjustifiably better rating.

4.2 Schematic Comparison of TruDiPhe to Other Reputation Systems

In the TDG, agents share computing resources with other agents. If successful interaction takes place (i.e. the worker returns the right result), the submitter creates a TruDiPhe with the trust value, timestamp, and the public key of the worker. The TruDiPhe is then sent to the worker (recipient) which may use the TruDiPhe to 'promote' itself. Of course, agents are free to store additional ratings, also negative ones, locally.

When an agent x is interested in the trust values of an agent y, it asks this agent to send all its relevant TruDiPhes (i.e. y is recipient and timestamp is in the right time frame set by the agent or the system protocol). These collections of TruDiPhes resemble the mixed scent profiles of ants: Although agents do not exchange their TruDiPhes, they can mutually add signatures to each other's profiles. The more agents in the network have interacted successfully with the queried agent, the more different signatures its profile contains, and hence, the more trustworthy it is. Note that TruDiPhes cannot actually mix because of the cryptographic encryption, but this is irrelevant for the algorithmic properties. Xiong & Liu (2004) found that a dynamical time frame (based on recent behaviour) outperforms a static one. This approach should therefore be chosen for systems with TruDiPhes.

In most systems, reputation managers exist, e.g. *Score Managers* in EigenTrust (Kamvar et al. (2003)), *Trust Managers* in PeerTrust (Xiong & Liu (2004)), or *Trust-Holding Agents* (THAs) in TrustMe (Singh & Liu (2003)), which maintain central reputation databases, mapped to the agents by hash functions. Figure 1 shows the schematic process of an agent b learning the reputation of an agent a in a system relying on consolidating agents: b looks up the reputation manager of a, which is c. Then, b sends a request for the reputation of a to c, which queries every node in the network (here: x, y and z) for their experiences with a. After receiving all replies, c calculates the reputation of a, and returns it to b.

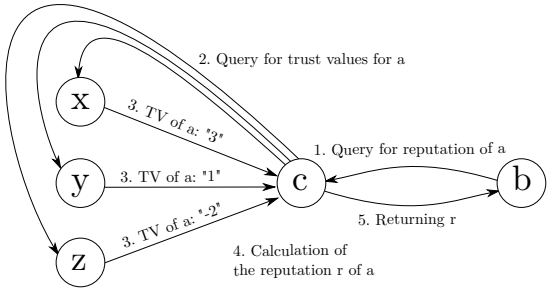

Figure 1 The schematic calculation of agent a's reputation in most systems with centralised trust value access schemes.

In our system (see Figure 2), trust values are queried directly from the recipients of trust values and are 'pushed' instead of 'pulled': After interacting with a, the agents x, y and z immediately send a TruDiPhe containing the trust value for that interaction to a. Agent b can now ask a for the TruDiPhes from a specific time frame (here: 1 to 3), b is then responsible for calculating the reputation of others by themselves.

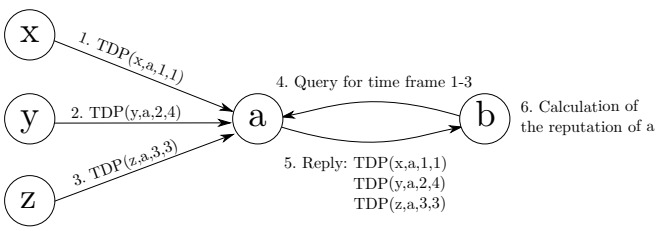

Figure 2 The schematic calculation of agent a's reputation with TruDiPhes.

With these definitions, we will now describe the characteristics of the TruDiPhe protocol.

4.3 Properties of the Protocol

A selection of properties (originally listed by Singh & Liu (2003) for their system *TrustMe*) for the concept of TruDiPhes is discussed in the following:

Anonymity and Accountability: Identities in TruDiPhe are anonymous in the sense that their real world counterparts do not need to be known to other members, although ratings can still be traced back to their creators.

Reliability: Cryptography secures that ratings in TruDiPhe cannot be manipulated. Dishonest ratings cannot be avoided in any system, however.

Persistence: Since the owner and the recipient of a TruDiPhe are the same, ratings are kept in the system for as long as they are needed. When an agent leaves the system, it takes all of the trust values with it which affect itself (except for direct experience ratings stored at its interaction partners). This is the ideal case regarding storage space.

Small decision time: Only one message is needed to find out the reputation of an agent in TruDiPhe, while other systems use multiple requests: at least two to three reputation managers need to be contacted to exclude system failures or dishonest ratings. If an agent is unavailable for a reputation request in TruDiPhe, it is also unavailable for work.

Ease of storing trust values: Again, only one message is required to contribute a trust value to the system.

5 Evaluation

As discussed above, TruDiPhe is a novel distributed approach to the dissemination of trust values. In this section, we demonstrate the effectiveness of this method. Furthermore, evaluating single components of the system enables the exact localisation of performance gains and losses. We already introduced the simulation environment and the agent types in Section 3. Initially, we formulate the research hypothesis that is investigated using the experimental evaluation. Afterwards, we verify that the hypothesis holds in terms of two experiments, before we discuss the results. We are not interested in sophisticated analysis of the application scenario, since we already compared the underlying concept against related work in Reif et al. (2016). Consequently, we simplify the particular trust ratings issued by individual agents (without loss of generality) as follows: A successful interaction results in a single positive value of '1'. This is encoded as follows: $P_{submitter}(B_{worker}, 1, TS, GenericType), B_{submitter}$. Compared to previous work, this does not affect the trust mechanism itself. In turn, it simplifies the trust-analysis since we do not have to distinguish between different kinds of ratings and the implications on the interpretation process.

We define the following hypothesis as basis for our experimental evaluation:

> *If a certain percentage of agents in a TDG is dishonest, reputation systems using TruDiPhes outperform comparable systems with reputation managers.*

5.1 Experimental Setup

The simulation runs in discrete time steps (ticks). Processing one work unit (WU) requires 750 to 1000 ticks, one job is composed of 7 to 15 WUs. The simulation runs 300000 ticks for each experiment with 100 agents, whereby an agent can distribute only one work unit in one tick. If a WU cannot be passed on for the third time, e.g. due to false computation results or rejection, the agent processes the WU itself. Overhead in terms of bandwidth and messaging costs are neglected in the simulation results, since both are comparable to the standard mechanism of the TDG. For all other simulation parameters, we used the finding of the parameter study presented in Klejnowski (2014). We compare the TruDiPhe-implementation to the same implementation with two differences:

1. Negative trust ratings are allowed (for TruDiPhe, all negative ratings are set to 0)
2. We assume the "worst case", i.e the reputation manager as well as the dishonest agents in the system are collaborating by inverting all trust ratings for a maximum falsifying impact.

These two differences capture the workings of traditional reputation manager systems, as described in Section 2, and is referred to as "Others" in the figures.

5.2 Experimental Results

In the first experiment, we simulate a system with a fixed amount of EGO and a varying amount of DADA. At the beginning of the simulation, there are 80 ADA and 20 EGO. In steps of 5% (of the initial 80 ADA), DADA are incorporated into the system, replacing the respective amount of ADA. In other systems, the higher percentage of dishonest *ratings* is translated into a higher percentage of dishonest *agents*. This is reasonable, since for the final trust value it does not matter whether the value was handled by one or two agents in the system, as long as their probability to be dishonest is the same. The results of the simulation runs are depicted in Figure 3.

In TruDiPhe, the ADA can keep their performance level relatively stable, independent from how many dishonest agents participate in the system. DADA which are being added to the system can only bring limited damage to TruDiPhe, as they can rate ADA with 0 as a minimum. This means that they are not able to pull down the reputation values as much as in the other systems where ratings lower than 0 are allowed. ADA keep a reputation comparable to the EGO and, in consequence, are chosen to process WUs more often than in the other systems—where a higher percentage

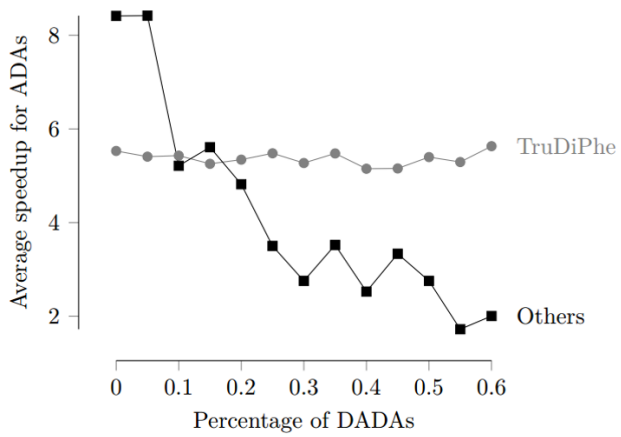

Figure 3 Average speedup of ADAs for different percentages of DADAs and a fixed amount of EGOs in 'traditional' systems (with reputation managers) compared to a simulation with TruDiPhe with a total of 100 agents.

of dishonest ratings results in a clearly worse and decreasing performance.

In the second experiment, we combine DEGO and dishonest reputation managers. We start with 100 ADA and introduce DEGO in 5% steps. As dishonest reputation managers can only change reputation values and do not have any other harmful affect on the system, the same amount of DEGO as in TruDiPhe is used, while the other dishonest agents are made up of DADA. The results are shown in Figure 4.

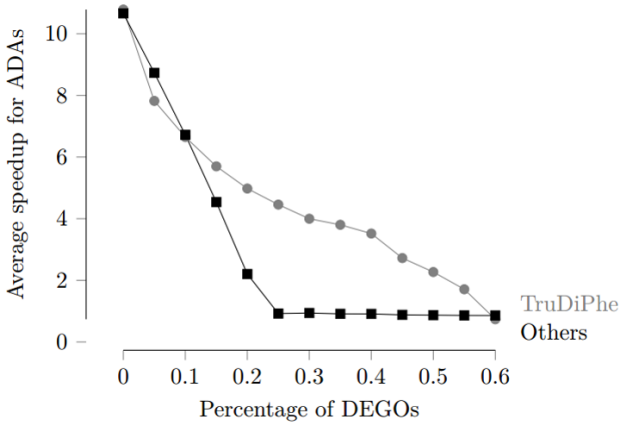

Figure 4 Average speedup of ADAs for different percentages of DEGOs in traditional systems with reputation managers and in TruDiPhe (100 agents).

As we can see, TruDiPhe outperforms the other systems from about 11% DEGO on. If more than a quarter of agents is dishonest in the case of other systems, the benefit of participating in the system is worse than processing the jobs by oneself. It should be noted that this, as well as the previous Figure 3, depicts the worst case of dishonest behaviour. The ratings are inverted in every case, which is improbable for systems in which dishonest reputation managers and dishonest trust creators do not cooperate closely with each other. Rather, one could assume that dishonest reputation managers sometimes reverse false ratings of other dishonest agents, changing them back to their original and correct value.

Systems with predominantly honest agents, such as traditional systems like TrustMe, fare better than systems with TruDiPhe. This is mainly due to the limitation of TruDiPhe to positive ratings. As soon as (dishonest) agents start to lie about trust values, systems with only positive ratings have an advantage. Furthermore, TruDiPhe has the additional benefit that no intermediaries (in terms of reputation managers) are employed which could manipulate the trust values. The correctness of the values is ensured by asymmetric encryption. When taking an increased probability of dishonest ratings into account, systems with traditional reputation managers give a lot less benefit to the honest and benevolent participants of the system than systems with TruDiPhes.

These considerations affirm the statement of our hypothesis. The "certain percentage" of dishonest agents is between 10 % and 20 % in our experiments, but is generally dependent on the exact reputation system using TruDiPhes, the amount of agents in the system, and other participating agent types

6 Conclusion

In this article, we presented a novel approach to ensure a fully distributed reputation system for open, heterogeneous, trusted Multi-agent Systems. This approach, called 'TruDiPhe', is inspired by concepts found in nature, more precisely in the chemical communication and nestmate recognition observed in ant colonies. TruDiPhe replaces a central reputation database or the need for reputation managers by storing trust values decentrally at the receiving agents. With the concept of asymmetric encryption, a TruDiPhe message ensures important aspects such as counterfeit security and copy protection.

The application scenario *Trusted Desktop Grid* is a simulation of an open Multi-agent System. In the presented evaluations, we compared the TDG with its reputation managers (as placeholder for comparable other systems) to the TDG with TruDiPhe applied. The results showed that the extension with TruDiPhe outperforms other systems as soon as there are more than about 10% to 20% malicious agents.

For future work, we want to answer more questions on the concept of TruDiPhe and distributed verification in general. Examples are: Is TruDiPhe able to counter more security threats, e.g. whitewashing or sybil attacks? If not, is it extendable in an apropriate way? How do systems with transitive trust fare against systems without in the face of dishonest agents?

References

Aberer, K., Despotovic, Z., Galuba, W. & Kellerer, W. (2006), The complex facets of reputation and trust, *in* 'Computational Intelligence, Theory and Applications', Springer, pp. 281–294.

Anderson, D. & Fedak, G. (2006), The Computational and Storage Potential of Volunteer Computing, *in* 'Proc. of CCGRID 2006', IEEE, pp. 73–80.

Anglano, C., Canonico, M., Guazzone, M., Botta, M., Rabellino, S., Arena, S. & Girardi, G. (2008), 'Peer-to-Peer Desktop Grids in the Real World: The ShareGrid Project', *Proc. of CCGrid 2008* pp. 609–614.

Bagnères, A.-G. & Lorenzi, M. C. (2010), Chemical deception/mimicry using cuticular hydrocarbons, *in* G. J. Blomquist & A.-G. Bagnres, eds, 'Insect Hydrocarbons', Cambridge University Press, pp. 282–324. Cambridge Books Online.

Bernard, Y., Klejnowski, L., Hähner, J. & Müller-Schloer, C. (2010), Towards trust in desktop grid systems, *in* '10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGrid 2010, 17-20 May 2010, Melbourne, Victoria, Australia', pp. 637–642.

Carbo, J., Molina, J. M. & Davila, J. (2002), Comparing predictions of sporas vs. a fuzzy reputation system, *in* '3rd International Conference on Fuzzy Sets and Fuzzy Systems', Vol. 200, Citeseer, pp. 147–153.

Carbo, J., Molina, J. M. & Davila, J. (2003), 'Trust management through fuzzy reputation', *International Journal of Cooperative Information Systems* **12**(01), 135–155.

Carrara, E. & Hogben, G. (2007), 'Enisa position paper no. 2 reputation-based systems: a security analysis'.

Castelfranchi, C. & Falcone, R. (2010), *Trust Theory: A Socio-Cognitive and Computational Model*, Vol. 18, John Wiley & Sons.

Chakravarti, A. J., Baumgartner, G. & Lauria, M. (2004), Application-Specific Scheduling for the Organic Grid, *in* 'Proc. of GRID 2004 Workshops', IEEE, Washington, DC, USA, pp. 146–155.

Choi, S., Buyya, R., Kim, H. & Byun, E. (2008), A taxonomy of desktop grids and its mapping to state of the art systems, Technical report, Grid Computing and Dist. Sys. Laboratory, The University of Melbourne.

d'Ettorre, P. & Lenoir, A. (2010), Nestmate recognition, *in* 'Ant Ecology', Oxford University Press, pp. 109–194.

Dickinson, M. H. (1999), 'Bionics: Biological Insight Into Mechanical Design', *Proceedings of the National Academy of Sciences* **96**(25), 14208–14209.

Edenhofer, S., Stifter, C., Jänen, U., Kantert, J., Tomforde, S., Hähner, J. & Müller-Schloer, C. (2015), An accusation-based strategy to handle undesirable behaviour in multi-agent systems, *in* '2015 IEEE International Conference on Autonomic Computing, Grenoble, France, July 7-10, 2015', pp. 243–248.

Fedak, G., Germain, C., Neri, V. & Cappello, F. (2001), Xtremweb: A Generic Global Computing System, *in* 'Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on', pp. 582–587.

Greengard, S. (2015), *The Internet of Things*, MIT Press Essential Knowledge, MIT Press. ISBN-13: 978-0262527736.

Hölldobler, B. & Wilson, E. O. (2009), *The Superorganism: The Beauty, Elegance, and Strangeness of Insect Societies*, W. W. Norton and Company.

Kamvar, S. D., Schlosser, M. T. & Garcia-Molina, H. (2003), The eigentrust algorithm for reputation management in p2p networks, *in* 'Proceedings of the 12th international conference on World Wide Web', ACM, pp. 640–651.

Kantert, J., Edenhofer, S., Tomforde, S. & Müller-Schloer, C. (2015), Representation of Trust and Reputation in Self-Managed Computing Systems, *in* 'IEEE 13th International Conference on Dependable, Autonomic and Secure Computing, DASC 2015', IEEE, Liverpool, UK, pp. 1827–1834.

Klejnowski, L. (2014), Trusted Community: A Novel Multiagent Organisation for Open Distributed Systems, PhD thesis, University of Hannover.

Leonhardt, S. D., Menzel, F., Nehring, V. & Schmitt, T. (2016), 'Ecology and evolution of communication in social insects', *Cell* **164**(6), 1277–1287.

Luhmann, N. (1979), *Trust and Power*, Chichester: Wiley.

Mnif, M. & Müller-Schloer, C. (2006), Quantitative emergence, *in* 'Proceedings of the 2006 IEEE Mountain Workshop on Adaptive and Learning Systems (SMCals 2006), held 24 Jul - 26 Jul 2006, Utah State University College of Engineering Logan, UT, USA', IEEE, Piscataway, NJ, USA, pp. 78–84.

Morgan, E. D. (2009), 'Trail pheromones of ants', *Physiological Entomology* **34**(1), 1–17.

Mui, L. (2002), Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks, PhD thesis, Massachusetts Institute of Technology.

Müller-Schloer, C., Schmeck, H. & Ungerer, T. (2011), *Organic ComputingA Paradigm Shift for Complex Systems*, Vol. 1, Springer Science & Business Media.

Ramchurn, S. D., Huynh, D. & Jennings, N. R. (2004), 'Trust in Multi-agent Systems', *The Knowledge Engineering Review* **19**(01), 1–25.

Reif, W., Anders, G., Seebach, H., Steghöfer, J.-P., Andre, E., Hähner, J., Müller-Schloer, C. & Ungerer, T. (2016), *Trustworthy Open Self-Organising Systems*, Springer. To appear.

Rouse, M. (2016), 'Public-key encrpytion'. Last accessed: 2016-02-10.

Sabater, J. & Sierra, C. (2005), 'Review on computational trust and reputation models', *Artificial Intelligence Review* **24**(1), 33–60.

Singh, A. & Liu, L. (2003), Trustme: Anonymous management of trust relationships in decentralized p2p systems, *in* 'Peer-to-Peer Computing, 2003.(P2P 2003). Proceedings. Third International Conference on', IEEE, pp. 142–149.

Tomforde, S., Hähner, J., Seebach, H., Reif, W., Sick, B., Wacker, A. & Scholtes, I. (2014), Engineering and Mastering Interwoven Systems, *in* 'ARCS 2014 - 27th International Conference on Architecture of Computing Systems, Workshop Proceedings, February 25-28, 2014, Luebeck, Germany, University of Luebeck, Institute of Computer Engineering', pp. 1–8.

Vu, H. L. (2010), High Quality P2P Service Provisioning via Decentralized Trust Management, PhD thesis, École Polytechnique Fédérale de Lausanne.

Wang, Y. & Vassileva, J. (2004), Trust-Based Community Formation in Peer-to-Peer File Sharing Networks, *in* 'Proc. on Web Intelligence', pp. 341–348.

Weiser, M. (1991), 'The computer for the 21st century', *Scientific American* **265**(3), 66–75.

Wooldridge, M. J. (2001), *Introduction to Multiagent Systems*, John Wiley & Sons, Inc., New York, NY, USA.

Xiong, L. & Liu, L. (2004), 'Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities', *Knowledge and Data Engineering, IEEE Transactions on* **16**(7), 843–857.

Zacharia, G. & Maes, P. (2000), 'Trust management through reputation mechanisms', *Applied Artificial Intelligence* **14**(9), 881–907.